Remarks

The Applicants have amended Claims 15 and 26. The Applicants have not cancelled or added any claims. Thus, Claims 15-30 remain pending.

Claims 15 and 26 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter regarded by the Applicants as the invention. In particular, the rejection states that there is insufficient antecedent basis for the term "the value." The Applicants have amended Claims 15 and 26 to recite "wherein a value of the parameter field corresponds to a parameter of said digital signals" to more clearly specify the meaning of the term "value." Thus, the Applicants submit that the claim language in Claims 15 and 26 is clear and distinct and accordingly requests withdrawal of the § 112 rejection.

Claims 15-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over US Patent Application Publication No. 2004/0054965 by Haskell et al. ("Haskell"). The Applicants respectfully request reconsideration and withdrawal of the rejection.

Haskell is directed to MPEG-4 browsers capable of facilitating the object-based flexibilities of MPEG-4 scenes. Fig. 2 and the associated discussion in paragraphs [33] and [37] of Haskell illustrate an MPEG-4 audio-visual object playback system. The system includes, in part, an MPEG-4 media decoder, a BIFS decoder, a compositor/renderer, and a display process. "The output of the media decoders of the MPEG-4 media decoder 230, as well as the output of the BIFS decoder 240 is provided to the compositor and renderer 250," the output of which is the presentation scene to be provided to the display process for display.

In paragraphs [48] and [49] of Haskell, as referenced in the rejection, data formatting of scene description in BIFS format is disclosed with reference to Figs. 8 and 9. In particular, a BIFS scene description graph interpreter, including nodes that are invoked for the contents of the scene

description graph, are described. The nodes deal with the content and attach the appropriate decoders for decoding the corresponding encoded portions of the data. Corresponding programmer interfaces are included between the nodes and the corresponding decoders.

Fig. 11 and paragraph [53] of Haskell teach direct interaction between a user and an MPEG audio-visual objects demultiplexer and BIFS browser and additional user interaction via use of scripting between the MPEG audio-visual objects demultiplexer and BIFS browser and the scene description graph interpreter.

Haskell, in Fig. 13 and paragraphs [63-65], discloses the addition of an adaptive audio-visual session (AAVS) module to transmit control data and information to the MPEG audio-visual objects demultiplexer and BIFS browser. An association between user interaction signals and objects is performed via this adaptation device that executes an AAVS external script containing updating information. Fig. 14 portrays the architectural details of Fig. 13 and is discussed in paragraphs [66-69] of Haskell. An external applet interface interfaces with an AAVS external script to provide external scripting to the MPEG audio-visual objects demultiplexer and BIFS browser. The external scripting may be from user interactions to program the behavior of the scene. The script may be, for example, written in Java.

However, the content of the "script" is not specified in Haskell, as is done in Claims 15 and 26. In particular, Haskell does not specify if a signal is built having the form of a BIFS node from user data and the script. While Haskell includes features that are used to manage user interactions requiring a high level of flexibility, as well as to dynamically change nodes from the scene, Haskell does not teach formatting user interaction data in order to have the form of a BIFS node. Haskell also does not teach the exact content of this signal in the form of a BIFS node. In particular, Claims 15 and 26 differ from the teachings of Haskell in that Claims 15 and 26 recite construction of a

digital sequence having the form of a BIFS node. The node of Claims 15 and 26 comprises "at least one field defining a type and number of interaction data to be applied to objects of said scene and said node specifying an association between digital signals of user interaction and scene objects." Haskell, in its teachings of user interaction with the MPEG objects and BIFS browser, does not teach a node with a field that defines the data to be applied to objects and that specifies an association. Instead, Haskell teaches simply of the possibility of user interactions of a scene. Furthermore, in Haskell's description of an external script in paragraphs [66-69], there is absolutely no teaching of a parameter field with a value that "corresponds to a parameter of said digital signals received from the peripheral command device," as is recited in Claims 15 and 26.

Claims 15-30 are further rejected under 35 U.S.C. §102(e) as being anticipated by US Patent No. 7,149,770 to Kalva et al. ("Kalva"). The Applicants respectfully request reconsideration and withdrawal of the rejection.

Kalva is directed to client-server interactions via a communication system based on the MPEG-4 standard, thus enabling a dynamic evolution of a scene to modify depending upon interaction data. Streams, such as an MPEG-4 video stream, contains scene description information, audio-visual object data, or control information. User interaction with the scene may be an option depending upon the information contained in the scene description (Kalva, column 4, lines 17-50).

In column 5, under the section "MPEG-4 Scene Description," Kalva states that MPEG-4 "defines a bandwidth-efficient compressed representation called BIFS (Binary Format for Scenes)." Node coding is performed, with the fields of each node assuming default values. Fields with non-default values are specified. Fields of a node act as event sources, event sinks, or both. The linking of event source fields to event sink fields causes scene behavior and interactivity. In column 7, line 67 through column 8, line 15, Kalva describes the use of command descriptors to associate

commands with event sources within the nodes of a scene graph. The descriptor begins with a tag that identifies it as a descriptor, following by descriptor ID, followed by command ID. A length indication, number of ES-IDs, desired ES-IDs, and application parameters follow.

Thus, Kalva includes the use of command descriptors with order delivery nodes, or server supply routes in the description of the scene, to provide support to specific interactivity for the application.

However, the command descriptor of Kalva is different than the constructed digital sequence of independent Claims 15 and 26. The command descriptor of Kalva does not take the form of a BIFS node, as shown in Fig. 5 of Kalva, while Claims 15 and 26 recite that a first digital sequence has the form of a BIFS node. Kalva does not at all disclose this possibility. Thus, Kalva cannot disclose that the node comprises a "field defining a type and number of interaction data to be applied to objects of said scene and said node specifying an association between said digital signals of user interactions and the scene objects," as is required by Claims 15 and 26. Withdrawal of the rejection based on Kalva is respectfully requested.

Additionally, neither Haskell nor Kalva suggest composing a digital signal describing the actions to be performed under the claimed format, with every action including a "nature of the action" field and "parameter for the action" fields in order to form a BIFS node. One skilled in the art, although knowing the composition of a sequence BIFS by nodes, would not have been led to achieve such nodes from data of user interaction defining actions to be applied to objects of scenes. Indeed, nothing suggests such nodes from these data, since only nodes from objects of scenes are known from the art and nothing suggests applying these nodes to actions as no advantageous result is expected.

The Applicants respectfully submit that the above differences set forth with respect to each of

Haskell and Kalva are such that Haskell and Kalva fail to result in a method or apparatus that contains each and every claimed aspect of the subject matter recited in independent Claims 15 and 26 and their respective dependent Claims 16-25 and 27-30. Thus, the Applicants respectfully submit that the claims are allowable over Haskell and Kalva.

In light of the foregoing, the Applicants respectfully submit that Claims 15-30 are now in condition for allowance, which is respectfully requested.

Respectfully submitted,

T. Daniel Christenbury
Reg. No. 31,750
Attorney for Applicants

TDC/EEP
(215) 656-3381